# Chapter 22
# Configure Tunnel Interfaces

By encapsulating arbitrary packets inside a transport protocol, tunneling provides a private, secure path through an otherwise public network. Tunnels connect discontinuous subnetworks and enable encryption interfaces, virtual private networks (VPNs), and Multiprotocol Label Switching (MPLS). If you have a Tunnel PIC installed in your router, you can configure unicast and multicast tunnels.

You can configure two types of tunnels for VPNs: one to facilitate route table lookups and another to facilitate VPN routing and forwarding instance (VRF) table lookups.

For information about encryption interfaces, see “Configure Encryption Interfaces” on page 223 and the *JUNOS Internet Software Configuration Guide: Getting Started*. For information about VPNs, see the *JUNOS Internet Software Configuration Guide: VPNs*. For information about MPLS, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

The JUNOS software supports the following tunnel encapsulations:

- Generic route encapsulation (GRE)
- IP over IP (IP-IP)
- Virtual Private Network (VPN)
- PIM encapsulation

This chapter discusses the following tasks you can perform to configure tunnel interfaces:

- Configure a Unicast Tunnel on page 312
- Configure a Multicast Tunnel on page 313
- Configure a VPN Loopback Tunnel for Route Table Lookup on page 313
- Configure a VPN Loopback Tunnel for VRF Table Lookup on page 314
- Configure PIM Tunnels on page 315
- Configure an IPv6 over IPv4 Tunnel on page 315
- Example 1: Configure Unicast Tunnels on page 316
- Example 2: Configure a VPN Loopback Tunnel for VRF Table Lookup on page 317
- Example 3: Configure an IPv6 over IPv4 Tunnel on page 318

## Configure a Unicast Tunnel

To configure a unicast tunnel, you configure the gr interface (to use GRE encapsulation) or the ip interface (to use IP-IP encapsulation) and include the tunnel statement:

```
[edit interfaces]
gr-fpc/pic/port or ip-fpc/pic/port {
    unit logical-unit-number {
        tunnel {
            source address;
            destination address;
            routing-instance {
                destination routing-instance-name;
            }
            ttl number;
        }
        family family {
            address address {
                destination address;
            }
        }
    }
}
```

You can configure multiple logical units for each GRE or IP-IP interface, and you can configure only one tunnel per unit.

Each tunnel interface must be a point-to-point interface. Point to point is the default interface connection type, so you do not need to include the point-to-point statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.

You must specify the tunnel's destination and source addresses. The remaining statements are optional.

To set the TTL field that is included in the encapsulating header, include the ttl statement. If you explicitly configure a TTL value for the tunnel, you must configure it to be one larger than the number of hops in the tunnel. For example, if the tunnel has seven hops, you must configure a TTL value of 8.

You must configure at least one family on the logical interface. To enable MPLS over GRE tunnel interfaces, you must include the family mpls statement in the GRE interface configuration. In addition, you must configure the protocols statements to enable RSVP, MPLS, and LSPs over GRE tunnels; for more information, see "Example 1: Configure Unicast Tunnels" on page 316 and the *JUNOS Internet Softw are Configur ation Guide: MPLS Applications*.

Unicast tunnels are bidirectional.

A configured tunnel cannot go through Network Address Translation (NAT) at any point along the way to the destination.

## Configure a Multicast Tunnel

For interfaces that carry IPv4 or IPv6 traffic, you can configure multicast tunnels. To configure a multicast tunnel, include the multicasts-only statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] or [edit interfaces *interface-name* unit *logical-unit-number* family inet6] hierarchy level:

```
multicasts-only;
```

Multicast tunnels filter all unicast packets; if an incoming packet is not destined for a 224/8 or greater prefix, the packet is dropped and a counter is incremented.

You can configure this property on GRE, IP-IP, PIM, and MT tunnels only.

## Configure a VPN Loopback Tunnel for Route Table Lookup

To configure tunnel interfaces to facilitate route table lookups for VPNs, you specify a tunnel's endpoint IP addresses and associate them with a routing instance that belongs to a particular routing table. This enables the software to search in the appropriate routing table for the route prefix, because the same prefix can appear in multiple routing tables. To configure the destination VPN, include the routing-instance statement at the [edit interfaces gr-*fpc/pic/port* unit *logical-unit-number* tunnel] hierarchy level:

```
[edit interfaces]
gr-fpc/pic/port {
    unit logical-unit-number {
        tunnel {
            source address;
            destination address;
            routing-instance {
                destination routing-instance-name;
            }
        }
    }
}
```

This configuration indicates that the tunnel's destination address is in routing instance *routing-instance-name*. By default, the tunnel route prefixes are assumed to be in the default Internet routing table inet.0.

> **Note**
>
> You can configure a VPN loopback tunnel for either VRF table lookup or route table lookup, not both. For more information, see "Configure a VPN Loopback Tunnel for VRF Table Lookup" on page 314.

For more information about VPNs, see the *JUNOS Internet Softw are Configur ation Guide: VPNs*.

## Configure a VPN Loopback Tunnel for VRF Table Lookup

You can configure a VPN loopback tunnel to facilitate VPN routing and forwarding instance (VRF) table lookup based on MPLS labels. You might want to enable this functionality so you can do either of the following:

1. Forward traffic on a PE-router-to-CE-device interface, in a shared medium, where the CE device is a Layer 2 switch without IP capabilities (for example, a metro Ethernet switch).

   The first lookup is done based on the VPN label to determine which VRF table to refer to, and the second lookup is done on the IP header to determine how to forward packets to the correct end hosts on the shared medium.

2. Perform egress filtering at the egress PE router.

   The first lookup on the VPN label is done to determine which VRF table to refer to, and the second lookup is done on the IP header to determine how to filter and forward packets. You can enable this functionality by configuring output filters on the VRF interfaces.

To configure a tunnel interface to facilitate VPN routing and forwarding (VRF) table lookup based on MPLS labels, you specify a VPN loopback tunnel interface name and associate it with a routing instance that belongs to a particular routing table. The VPN loopback tunnel loops the packet back through the tunnel interface for route lookup. To specify a VPN loopback tunnel interface name, you configure the vt interface and include the family inet and family mpls statements:

```
[edit interfaces]
vt-fpc/pic/port {
    unit 0 {
        family inet;
        family mpls;
    }
    unit 1 {
        family inet;
    }
}
```

To associate the VPN loopback tunnel with a routing instance, include the VPN loopback tunnel interface name at the [edit routing-instances] hierarchy level:

```
[edit routing-instances] {
interface vt-fpc/pic/port;
```

> **Note** You can configure a VPN loopback tunnel for either VRF table lookup or route table lookup, not both. For more information, see “Configure a VPN Loopback Tunnel for Route Table Lookup” on page 313.

For the vt interface, none of the statements in the tunnel configuration block are valid.

Another way to configure egress filtering is to include the vrf-table-label statement at the [edit routing-instances *instance-name*] hierarchy level. However, for core-facing interfaces, this feature works for four-port E1, four-port E3, and SONET PPP/HDLC interfaces only. There is no restriction on CE router to PE router interfaces. If you enable egress filtering by configuring the vt interface on routers equipped with a Tunnel PIC, there is no restriction on the type of core-facing interface used or CE router to PE router interface used. You cannot configure a vt interface and the vrf-table-label statement at the same time. For more information about the vrf-table-label statement, see the *JUNOS Internet Software Configuration Guide: VPNs* .

## Configure PIM Tunnels

PIM tunnels are enabled automatically on routers that have a tunnel PIC and on which you enable PIM sparse mode. You do not need to configure the tunnel interface.

PIM tunnels are unidirectional.

In PIM sparse mode, the first-hop router encapsulates packets destined for the Rendezvous Point (RP) router. The packets are encapsulated with a unicast header and are forwarded through a unicast tunnel to the RP. The RP then de-encapsulates the packets and transmits them through its multicast tree. To perform the encapsulation and de-encapsulation, the first-hop and RP routers must be equipped with Tunnel PICs.

The JUNOS Internet software creates two interfaces to handle PIM tunnels:

pe—Encapsulates packets destined for the RP. This interface is present on the first-hop router.

pd—De-encapsulates packets at the RP. This interface is present on the RP.

## Configure an IPv6 over IPv4 Tunnel

If you have a Tunnel PIC installed in your router, you can configure IPv6 over IPv4 tunnels. To do this, you configure a unicast tunnel across an existing IPv4 network infrastructure. IPv6 packets are encapsulated in IPv4 headers and sent across the IPv4 infrastructure through the configured tunnel. You manually configure configured tunnels on each end point.

Configured IPv6-over-IPv4 tunnels are defined in RFC 2893, *Transition Mechanisms for IPv6 Hosts and Routers*. For information about configuring a unicast tunnel, see “Configure a Unicast Tunnel” on page 312. For an IPv6 over IPv4 tunnel configuration example, see “Example 3: Configure an IPv6 over IPv4 Tunnel” on page 318.

## Example 1: Configure Unicast Tunnels

Configure two unnumbered IP-IP tunnels:

```
[edit]
interfaces
    ip-0/3/0 {
        unit 0 {
            tunnel {
                source 192.168.4.18;
                destination 192.168.4.253;
            }
            family inet;
            family iso;
        }
        unit 1 {
            tunnel {
                source 192.168.4.18;
                destination 192.168.4.254;
            }
            family inet;
            family iso;
        }
    }
}
```

To configure a numbered tunnel interface, include an address under family inet:

```
[edit]
interfaces
    ip-0/3/0 {
        unit 0 {
            tunnel {
                source 192.168.4.18;
                destination 192.168.4.253;
            }
            family inet {
                address 5.5.5.1/30;
            }
            family iso;
        }
        unit 1 {
            tunnel {
                source 192.168.4.18;
                destination 192.168.4.254;
            }
            family inet {
                address 6.6.6.100/30;
            }
            family iso;
        }
    }
}
```

To configure MPLS over GRE tunnels, include the family mpls statement:

```
interfaces {
    gr-1/2/0 {
        unit 0 {
            tunnel {
                source 192.168.1.1;
                destination 192.168.1.2;
            }
            family inet {
                address 5.1.1.1/30;
            }
            family iso;
            family mpls;
        }
    }
}
```

## Example 2: Configure a VPN Loopback Tunnel for VRF Table Lookup

Configure a VPN loopback tunnel for VRF table lookup:

```
[edit routing-instances]
routing-instance-1 {
    instance-type vrf;
    interface vt-1/0/0.0;
    interface so-0/2/2.0;
    route-distinguisher 2:3;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    routing-options {
        static {
            route 10.0.0.0/8 next-hop so-0/2/2.0;
        }
    }
}
routing-instance-2 {
    instance-type vrf;
    interface vt-1/0/0.1;
    interface so-0/3/2.0;
    route-distinguisher 4:5;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
    routing-options {
        static {
            route 10.0.0.0/8 next-hop so-0/3/2.0;
        }
    }
}

[edit interfaces]
vt-1/0/0 {
    unit 0 {
        family inet;
        family mpls;
    }
    unit 1 {
        family inet;
    }
}
```

## Example 3: Configure an IPv6 over IPv4 Tunnel

Configure a tunnel on both sides of the connection.

On Router 1:

```
[edit]
interfaces {
   gr-1/0/0 {
       unit 0 {
          tunnel {
             source 10.19.2.1;
             destination 10.19.3.1;
          }
          family inet6 {
             address 7019::1/126;
          }
       }
    }
}
```

On Router 2:

```
[edit]
interfaces {
   gr-1/0/0 {
       unit 0 {
          tunnel {
             source 10.19.3.1;
             destination 10.19.2.1;
          }
          family inet6 {
             address 7019::2/126;
          }
       }
   }
}
```